46

UCID- 21866

ANTIDIFFUSIVE VELOCITIES FOR MULTIPASS DONOR CELL ADVECTION

L.G. Margolin
Lawrence Livermore National Laboratory
Livermore, CA 94550

Piotr K. Smolarkiewicz
National Center for Atmospheric Research
Boulder, CO 80307

December 1989

Lawrence Livermore National Laboratory

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

Price Code	Page Range
A01	Microfiche
Papercopy Prices	
A02	001-050
A03	051-100
A04	101-200
A05	201-300
A06	301-400
A07	401-500
A08	501-600
A09	601

ANTIDIFFUSIVE VELOCITIES
FOR
MULTIPASS DONOR CELL ADVECTION

L.G. Margolin
Lawrence Livermore National Laboratory
Livermore, CA 94550

Piotr K. Smolarkiewicz
National Center for Atmospheric Research
Boulder, CO 80307

ABSTRACT

Smolarkiewicz [1] describes an iterative process for approximating the advection equation. Basically, he uses a donor cell approximation to correct for the truncation error of the originally specified donor cell scheme. This step may be repeated an arbitrary number of times, leading to successively more accurate solutions to the advection equation. In this report, we show how to sum the successive approximations analytically to find a single antidiffusive velocity that represents the effects of an arbitrary number of passes. The analysis is first done in one dimension to illustrate the method. The analysis is then repeated in two dimensions. The existence of cross terms in the truncation analysis of the two-dimensional equations introduces an extra complication into the calculation. We discuss the implementation of our new antidiffusive velocities and provide some examples of applications.

1. INTRODUCTION

Smolarkiewicz [1] describes an iterative process for approximating the advection equation. MPDATA (Multidimensional Positive Definite Advection Transport Algorithm) is second-order accurate, positive definite,[†] conservative, and computationally efficient. MPDATA is iterative in nature. The first pass is a simple donor cell approximation, sometimes called upstream differencing, which is positive definite but only first-order accurate. The second pass increases the accuracy of the calculation by estimating and compensating the truncation error of the first pass. Additional

[†] Indeed MPDATA is sign-preserving [2,Sec.4]. However for historical reasons we shall refer to this property as positive-definiteness.

passes can be used to estimate the residual error of the previous pass and approximately compensate for it. This step may be repeated an arbitrary number of times, leading to successively more accurate solutions of the advection equation.

The basic idea is to use the positive definite properties of donor cell schemes to estimate the residual truncation error. Thus MPDATA consists of a sequence of donor cell steps. In the first pass, the velocity is the physical velocity. In the second and subsequent passes, the velocity is calculated from the field that is being advected and has no physical significance. These velocities are termed antidiffusive, or equivalently pseudo velocities.

In this report, we show how to analytically sum the successive approximations to find a single antidiffusive velocity that represents the effects of an arbitrary number of passes. From the truncation analysis of the donor cell scheme, we show that the antidiffusive velocities obey a recursion relation in which the velocity for any pass is simply related to the velocity of the previous pass. We then show how to sum the velocities *approximately* as a series in a small variable. The analysis is first done in one dimension in Sec. 3 to illustrate the method. The analysis is then repeated in two dimensions in Sec. 6. The existence of cross terms in two dimensions introduces some complications that are discussed in Sec. 5. We shall refer to the analytic sums of the antidiffusive velocities as recursive velocities, because they are derived from recursion relations.

With the new analytic results. MPDATA can be rewritten as a two pass scheme. It is still positive definite and second-order accurate. In contrast to the pseudo velocities of the original MPDATA [1], the recursive velocities do not necessarily satisfy the CFL conditions and must be appropriately bounded to insure stability. We will discuss this point in detail in Sec. 7. We also note that the possibility of summing the iteration velocities was mentioned briefly in [1,Sec. 5.5] but never explored.

The analytic formulae for the recursive velocities are more complicated than for the iterative velocities. Rough comparisons in Sec. 8 show that the computer time required for the new scheme is comparable to that of the original MPDATA with three total iterations (i.e., the original donor cell calculation plus two corrective passes). Our calculations also show that the old scheme reaches comparable accuracy with the new only after four or more iterations. However this admittedly small saving of CPU time is not the principal reason for this work. Much bigger savings can be realized when MPDATA is used in conjunction with other enhancements such as a nonoscillatory option [3] which enforces strict monotonicity of the solution, or with third-order corrections which result in a third-order accurate version of MPDATA. In general, as we increase the total amount of work done in each iteration, the effect of the more complex pseudo velocities becomes less important and savings up to 50% become possible. Also the recursion velocities may provide an attractive option for those dynamic programs whose architecture does not permit numerous passes. Finally, we believe that the mathematic techniques described in this report are interesting in an abstract sense and lead to a deeper understanding of the MPDATA family of schemes.

2. TRUNCATION ANALYSIS

The advection equation in one dimension is

$$\frac{\partial \Psi}{\partial t} = - \frac{\partial}{\partial x} (u\Psi) \tag{2.1}$$

where the velocity u may vary in space and time. The donor cell or upstream approximation to the advection equation is written in flux form

$$\Psi^{n+1}_{i+1/2} = \Psi^{n}_{i+1/2} - [\, F(\Psi^{n}_{i+1/2}, \Psi^{n}_{i+3/2}, U_{i+1}) - F(\Psi^{n}_{i-1/2}, \Psi^{n}_{i+1/2}, U_{i}) \,] \tag{2.2}$$

where the flux function F is defined in terms of the local Courant number U by

$$\begin{gathered} F(\Psi_L, \Psi_R, U) \equiv 0.5 [\, (U + |U|) \Psi_L + (U - |U|) \Psi_R] \\ U \equiv \frac{u\, \delta t}{\delta x} \end{gathered} \tag{2.3}$$

The half integer indices correspond to the cell centers as illustrated in Fig. 2.1. Here δt is the computational time step and δx is the length of a cell.

Assume for the sake of simplicity of further discussion that the velocity is uniform in space. A simple truncation analysis, expanding about the time level n and spatial point (j+1/2), shows that (2.2) more accurately approximates the advection-diffusion equation

$$\frac{\partial \Psi}{\partial t} = - \frac{\partial}{\partial x} (u\Psi) + \frac{\partial}{\partial x} \left(K \frac{\partial \Psi}{\partial x}\right) \tag{2.4}$$

where

$$K = \frac{(\delta x)^2}{2\, \delta t} (|U| - U^2) \tag{2.5}$$

Thus (2.2) approximates the solution to the advection equation with a second-order error -- i.e., $O(\delta x^2)$. To improve the accuracy, it is necessary to construct a numerical estimate of the error and subtract it from (2.2).

The donor cell scheme (2.2) is positive definite for any velocity field and is monotone if the velocity field is constant in space, providing that the Courant number is properly bounded. These properties are lost in any scheme that is a linear combination of donor cell and centered differencing (c.f. [4,5]).

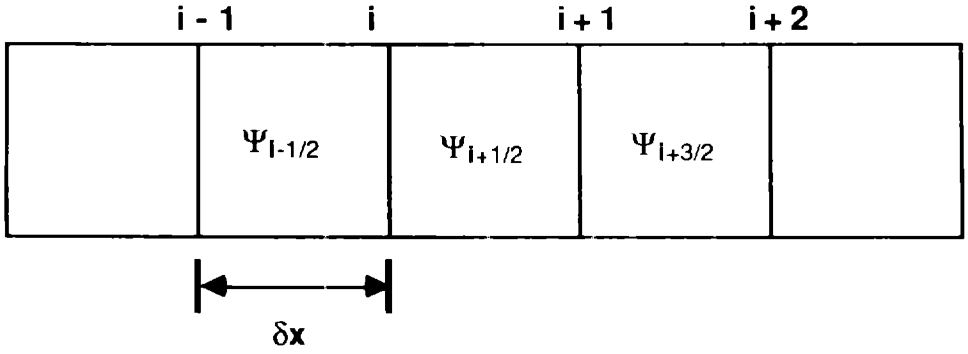

Figure 2.1: The cell centers are designated by half integer indices and the edges by full integers.

In these terms, Smolarkiewicz' idea can be stated very simply -- use a donor cell approximation to the error term. Since the error term is not written in a form to do this directly, he first rewrites the error term as

$$(\text{error})^{(1)} = \frac{\partial}{\partial x}\left(v^{(1)}\Psi\right) \qquad (2.6)$$

where

$$v^{(1)} \equiv \frac{(\delta x)^2}{2\,\delta t}\left(|U| - U^2\right)\frac{1}{\Psi}\frac{\partial \Psi}{\partial x} \qquad (2.7)$$

is a pseudo velocity. The superscript $^{(1)}$ has been added to show that it is the first approximation to subtracting the error. Inside the derivative in (2.6), the diffusive flux in the second term of (2.4) is multiplied by a factor of Ψ over Ψ --i.e., by unity. However, in the donor cell approximation to (2.6), the factor in the numerator will be represented using an upstream value whereas the factor in the denominator will be approximated using a centered value. In this way, a nonlinearity is introduced into the approximation and a higher-order approximation is found that still preserves the positivity property.

To compensate for the error between the donor cell solution $\Psi^{(1)}$ and a second-order accurate solution Ψ^{n+1}, we need to use the error of (2.6) evaluated at time level n+1. A second-order accurate estimate of the pseudo velocity (nondimensionalized for convenience) is

$$V_i^{(1)} \equiv \left(|U| - U^2\right)\frac{\Psi^{(1)}_{i+1/2} - \Psi^{(1)}_{i-1/2}}{\Psi^{(1)}_{i+1/2} + \Psi^{(1)}_{i-1/2}} = \left(|U| - U^2\right)A_i^{(1)} \qquad (2.8)$$

where

$$V^{(1)} = \frac{v^{(1)}\,\delta t}{\delta x} \qquad (2.9)$$

and

$$A_i^{(1)} = \frac{\Psi^{(1)}_{i+1/2} - \Psi^{(1)}_{i-1/2}}{\Psi^{(1)}_{i+1/2} + \Psi^{(1)}_{i-1/2}} \qquad (2.10)$$

$\Psi^{(1)}$ is the solution after the first donor cell pass.

As a second pass, we subtract a donor cell estimate of the error to improve the order of the approximation. The equation of the second pass is

$$\begin{aligned}\Psi^{(2)}{}_{i+1/2} = \Psi^{(1)}{}_{i+1/2} - F(\Psi^{(1)}{}_{i+1/2}, \Psi^{(1)}{}_{i+3/2}, V^{(1)}{}_{i+1}) \\ + F(\Psi^{(1)}{}_{i-1/2}, \Psi^{(1)}{}_{i+1/2}, V^{(1)}{}_{i-1})\end{aligned} \qquad (2.11)$$

We now expand (2.11) in a Taylor series, and find the residual error after the second pass -- e.g., $(\text{error})^{(2)}$

$$(\text{error})^{(2)} = \frac{1}{2}\frac{\partial}{\partial x}[\,(|V^{(1)}| - (V^{(1)})^2)\frac{\partial\Psi}{\partial x}\,] \equiv \frac{1}{2}\frac{\partial}{\partial x}(V^{(2)}\Psi) \qquad (2.13)$$

where

$$(V)^{(2)} = [\,(|V^{(1)}| - (V^{(1)})^2)\frac{\delta x}{2\Psi}\frac{\partial\Psi}{\partial x}\,] = A\,(|V^{(1)}| - (V^{(1)})^2)$$

Now when we implement the corrections to find $\Psi^{(3)}$, we write

$$\begin{aligned}\Psi^{(3)}{}_{i+1/2} = \Psi^{(2)}{}_{i+1/2} - F(\Psi^{(1)}{}_{i+1/2}, \Psi^{(1)}{}_{i+3/2}, V^{(2)}{}_{i+1}) \\ + F(\Psi^{(1)}{}_{i-1/2}, \Psi^{(1)}{}_{i+1/2}, V^{(2)}{}_{i-1})\end{aligned} \qquad (2.14)$$

and we continue to use $A^{(1)}$. Note that within the flux functions in (2.14) we continue to use $\Psi^{(1)}$. It is crucial to our development that A can be treated as a constant through all subsequent iterations. This contrasts with the original MPDATA approach which would use $A^{(2)}$ and $\Psi^{(2)}$ in (2.13) and (2.14), respectively.

The entire process of estimating the residual error and compensating it can be continued, iteration after iteration, with the general result for the iteration velocity being

$$(V)^{(k)} = [\,(|V^{(k-1)}| - (V^{(k-1)})^2)\frac{\delta x}{2\Psi}\frac{\partial\Psi}{\partial x}\,] = A\,(|V^{(k-1)}| - (V^{(k-1)})^2) \qquad (2.15)$$

With this approach, the result of doing multiple passes is equivalent to a single corrective pass using the velocity W

$$W = \sum_{k=1}^{\infty} V^{(k)} \qquad (2.16)$$

It is W that we have named the recursive velocity in Sec. 1. The main purpose of this report is the analytic calculation of W.

Note that successive passes after the second in the original MPDATA [1] continue to reduce the magnitude of the error, but do not reduce the order of the error, which remains at third-order. Similarly, using the recursive velocity W as the second pass in a two pass scheme reduces the magnitude of the error, but not its order.

3. SUMMING THE RECURSION

The fundamental problem can now be stated. Given the recursion relation of (2.15),

$$(V)^{(k)} = A\left(\left|V^{(k-1)}\right| - (V^{(k-1)})^2\right)$$

and the initial condition in (2.8),

$$V^{(1)} \equiv (|C| - C^2)\, A$$

can the sum in (2.16) be evaluated? The answer is that the sum can not be evaluated exactly, but that it can be evaluated *approximately* as a perturbation series in a small variable that we will define shortly.

First, we note that the constant A from (2 10)

$$A_j = \frac{\Psi_{i+1/2} - \Psi_{i-1/2}}{\Psi_{i+1/2} + \Psi_{i-1/2}}$$

lies in the interval {-1, 1} whenever the field Ψ is positive. Further, when the field is of variable sign, Smolarkiewicz and Clark displace the entire field by adding a large positive constant, transporting the new field which is positive, and then subtracting out the original constant [2]. Thus we may take

$$-1 \leq A \leq 1 \tag{3.1}$$

without loss of generality.

Next, we note that

$$\begin{aligned} &[0 \leq A \leq 1] \text{ and } [\left| V^{(k-1)}\right| \leq 1] \Rightarrow [0 \leq V^{(k)} \leq V^{(k-1)} \leq 1] \\ &[0 \geq A \geq -1] \text{ and } [\left| V^{(k-1)}\right| \leq 1] \Rightarrow [0 \geq V^{(k)} \geq V^{(k-1)} \geq -1] \end{aligned} \tag{3.2}$$

Since $|V^{(1)}| < 1$, we can use induction to show that all the pseudo velocities in the recursion will have the same sign. Further, the sign of the pseudo velocities is the same as the sign of A. (3.2) actually is a stronger statement concerning the convergence of the series. By the ratio test [6], the inequalities of (3.2) proves the convergence of the sum for the total velocity W in (2.16).

We begin the analysis by expanding the total velocity W in a series in the small variable ξ

$$W = \sum_{j=1}^{\infty} w_{(j)}\, \xi^{j} \tag{3.3}$$

Next, we expand each of the individual pseudo velocities in its own series

$$V^{(k)} = \alpha^{(k)}\xi + \beta^{(k)}\xi^{2} + \gamma^{(k)}\xi^{3} + \delta^{(k)}\xi^{4} + \ldots \tag{3.4}$$

Now substitute the expansion (3.4) into the recursion relation (2.15). Since each $V^{(k)}$ has the same sign, which is the sign of A, we can write

$$\begin{aligned} V^{(k+1)} = \{|A|\,\alpha^{(k)}\}\xi + \{|A|\,\beta^{(k)} - A\,\alpha^{(k)}\alpha^{(k)}\}\,\xi^{2} + \\ \{|A|\,\gamma^{(k)} - 2\,A\,\alpha^{(k)}\beta^{(k)}\}\,\xi^{3} + \ldots \end{aligned} \tag{3.5}$$

Since there is also a series expansion for $V^{(k+1)}$ similar to (3.4), we can write a set of recursive relations for the various subseries by requiring each of the coefficients of the expansion variable ξ in both series to match

$$\begin{aligned} \alpha^{(k+1)} &= |A|\,\alpha^{(k)} \\ \beta^{(k+1)} &= \{|A|\,\beta^{(k)} - A\,\alpha^{(k)}\alpha^{(k)}\} \\ \gamma^{(k+1)} &= \{|A|\,\gamma^{(k)} - 2\,A\,\alpha^{(k)}\beta^{(k)}\} \\ \delta^{(k+1)} &= \{|A|\,\delta^{(k)} - 2\,A\,\alpha^{(k)}\gamma^{(k)} - A\,\beta^{(k)}\beta^{(k)}\} \end{aligned} \tag{3.6}$$

plus the initial conditions. The initial conditions will of course depend upon the choice of the expansion variable ξ. The simplest choice for ξ is simply $V^{(1)}$ itself -- i.e., see (2.8).

$$V^{(1)} \equiv \frac{\delta x}{2\,\Psi}\,(|U| - U^{2})\,\frac{\partial\Psi}{\partial x} \equiv A\,(|U| - U^{2})$$

Then $|\xi| < 0.25$ and the initial conditions have the simple form

$$\alpha^{(1)} = 1 \;\; ; \;\; \beta^{(1)} = 0 \;\; ; \;\; \gamma^{(1)} = 0 \;\; ; \;\; \delta^{(1)} = 0 \;\; ; \; \ldots . \tag{3.7}$$

It is easy to solve the individual recursion relations in (3.6). For example, summing the first relation of (3.6) over k=1,2,3, . . . yields

$$\sum_{k=1}^{\infty} \alpha^{(k+1)} = |A| \sum_{k=1}^{\infty} \alpha^{(k)} = \sum_{k=1}^{\infty} \alpha^{(k)} - \alpha^{(1)} \tag{3.8}$$

One can solve algebraically for the sum. With a slight compression of notation, we have

$$\Sigma_{\alpha} \equiv \sum_{k=1}^{\infty} \alpha^{(k)} = \frac{1}{1 - |A|} \tag{3.9}$$

Evaluation of the β-series requires an extra step. Again we begin by summing the second relation of (3.6) over k=1,2,3, . . .

$$\Sigma_{\beta} = |A|\, \Sigma_{\beta} - A\, \Sigma_{\alpha\alpha} \quad \text{where} \quad \Sigma_{\alpha\alpha} \equiv \sum_{k=1}^{\infty} \alpha^{(k)}\alpha^{(k)} \tag{3.10}$$

Now to evaluate the last sum, $\Sigma_{\alpha\alpha}$, we multiply the recursive relation for α (3.6) by itself and then sum. This yields

$$\Sigma_{\alpha\alpha} = \alpha^{(1)}\alpha^{(1)} + A^2\Sigma_{\alpha\alpha} \quad \text{or} \quad \Sigma_{\alpha\alpha} = \frac{1}{1 - A^2} \tag{3.11}$$

and so

$$\Sigma_{\beta} = \frac{-A}{(1 - A^2)\,(1 - |A|)} \tag{3.12}$$

The procedure illustrated above can be repeated for the γ-series, and so on. The fact that the recursive velocities all have the same sign is crucial to the analysis since it allows us to eliminate the absolute value signs in the recursion relation. Comparing (3.3) and (3.4), it is clear that

$$\begin{aligned} w_{(1)} &= \sum_{k=1}^{\infty} \alpha^{(k)} = \Sigma_{\alpha} \\ w_{(2)} &= \sum_{k=1}^{\infty} \beta^{(k)} = \Sigma_{\beta} \end{aligned} \tag{3.13}$$

and so on. Our final result is the total pseudo velocity, W, which we can write

$$W = [\Sigma_\alpha]\,\xi + [\Sigma_\beta]\,\xi^2 + [\Sigma_\gamma]\,\xi^3 + \dots \qquad (3.14)$$

The summation terms can be written in a form that works for A of either sign

$$\begin{aligned} \Sigma_\alpha &= \frac{1}{(1 - |A|)} \\ \Sigma_\beta &= \frac{-A}{(1 - |A|)(1 - A^2)} \\ \Sigma_\gamma &= \frac{2\,|A|^3}{(1 - |A|)(1 - A^2)(1 - |A|^3)} \end{aligned} \qquad (3.15)$$

The coefficients of the higher order terms in the power series (3.14) get more complicated and also become less important. The number of terms that one keeps will depend on the desired accuracy, which in turn depends on the value of the expansion parameter ξ. In Sec. 8, we show examples of how the recursive velocities work, and also provide some data on the relative cost of using the recursive velocities versus using extra iterations.

4. EXTENSION TO MULTIDIMENSIONAL CALCULATIONS

The extension of our results for the recursive velocities from one dimension to two dimensions contains a subtlety originating in the cross derivatives that appear in the truncation analysis. However, the extension from two dimensions to three is straightforward, and for simplicity we describe only the two-dimensional case. We show a two-dimensional mesh in Fig. 4.1. The field Ψ is stored at the cell centers, denoted by half-integer indices. The velocity components at stored at the cell edges and have one integer index and one half-integer index. The coordinate x, the logical index i and the velocity component u correspond to the horizontal direction whereas the coordinate y, the logical index j and the velocity component v all correspond to the vertical direction.

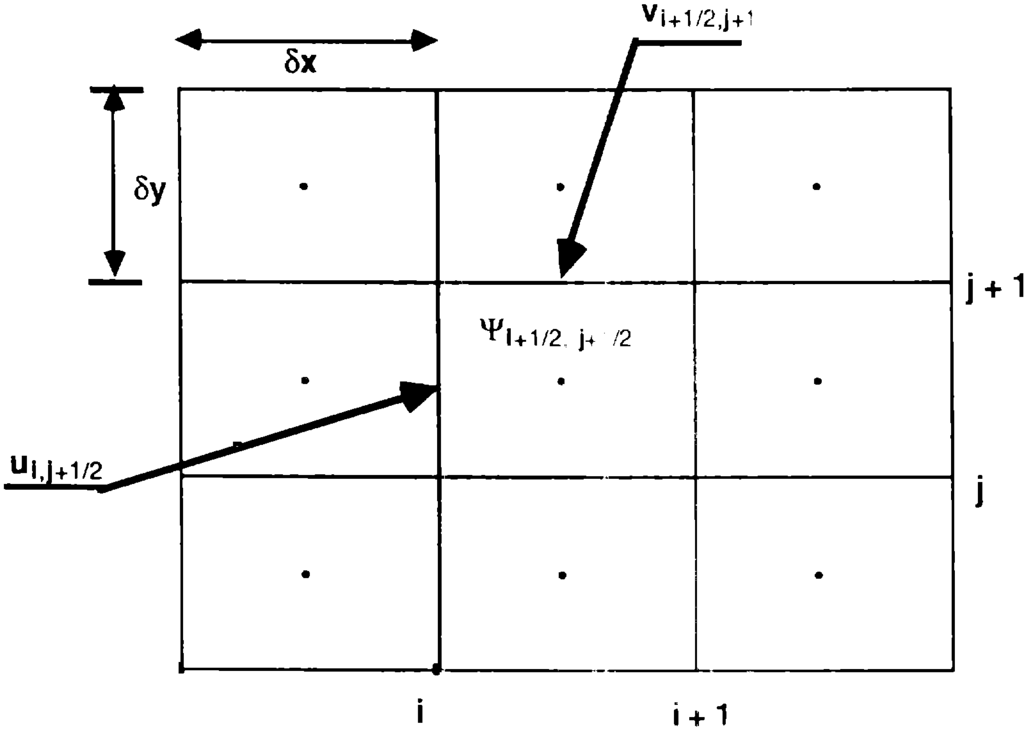

Figure 4.1: The cell centers are designated by half integer indices and the edges by full integers. The velocities are stored on the cell edges.

The numerical equation for donor cell in two dimensions can be written

$$\Psi^{n+1}_{i+1/2,j+1/2} = \Psi^{n}_{i+1/2,j+1/2} -$$

$$[\, F(\Psi^{n}_{i+1/2\,,j+1/2}, \Psi^{n}_{i+3/2,j+1/2}, U_{i+1\,,j+1/2}) - F(\Psi^{n}_{i-1/2\,,j+1/2}, \Psi^{n}_{i+1/2\,,j+1/2}, U_{i\,,j+1/2})\,] \quad (4.1)$$

$$[\, F(\Psi^{n}_{i+1/2\,,j+1/2}, \Psi^{n}_{i+1/2,j+3/2}, V_{i+1/2\,,j+1}) - F(\Psi^{n}_{i+1/2\,,j-1/2}, \Psi^{n}_{i+1/2\,,j+1/2}, V_{i+1/2,j})\,]$$

where now U and V are the dimensionless Courant numbers

$$U \equiv \frac{u\,\delta t}{\delta x} \quad \text{and} \quad V \equiv \frac{v\,\delta t}{\delta y} \qquad (4.2)$$

and the flux function F has been defined in (2.3).

Assuming that the velocities are constant in space, we expand (4.1) about the cell center (i+1/2,j+1/2) and time level n. The result is an advection-diffusion equation

$$\frac{\partial \Psi}{\partial t} = -\frac{\partial}{\partial x}(u\Psi) - \frac{\partial}{\partial y}(v\Psi) + \frac{|U|\,(\delta x)^2}{2\,\delta t}(\,1 - |U|\,)\frac{\partial^2 \Psi}{\partial x^2}$$

$$+ \frac{|V|\,(\delta y)^2}{2\,\delta t}(\,1 - |V|\,)\frac{\partial^2 \Psi}{\partial y^2} - \frac{U\,V\,(\delta x)(\delta y)}{\delta t}\,\frac{\partial^2 \Psi}{\partial x\,\partial y} \qquad (4.3)$$

The idea is now to write a donor cell estimate of the truncation error and subtract it from the difference equation (4.1) to create a higher-order approximation. Note that when we try to write this in a form using pseudo velocities, the cross-term gives us a degree of freedom not present in the one-dimensional analysis. That is, writing

$$\frac{\partial \Psi}{\partial t} = -\frac{\partial}{\partial x}(u\Psi) - \frac{\partial}{\partial y}(v\Psi) + \frac{\partial}{\partial x}(u^{(1)}\Psi) + \frac{\partial}{\partial y}(v^{(1)}\Psi) \qquad (4.4)$$

we can choose

$$u^{(1)} = \frac{|U|\,(\delta x)^2}{2\,\delta t}(\,1 - |U|\,)\frac{1}{\Psi}\frac{\partial \Psi}{\partial x} - (\,F\,)\frac{U\,V(\delta x)\,(\delta y)}{\delta t}\,\frac{1}{\Psi}\frac{\partial \Psi}{\partial y}$$

$$v^{(1)} = \frac{|V|\,(\delta y)^2}{2\,\delta t}(\,1 - |V|\,)\frac{1}{\Psi}\frac{\partial \Psi}{\partial y} - (\,1 - F)\frac{U\,V(\delta x)\,(\delta y)}{\delta t}\,\frac{1}{\Psi}\frac{\partial \Psi}{\partial x} \qquad (4.5)$$

where the real number F is arbitrary. We shall need this extra freedom when we sum the recursion.

The pseudo velocities must be evaluated at the cell edges. Their explicit form will depend upon whether the edge is horizontal or vertical. In either case we can write in dimensionless form,

$$U^{(1)} \equiv \frac{u^{(1)}\delta t}{\delta x} = |U|\,(1-|U|)\,A - 2\,(F)\,U\,V\,B \qquad (4.6)$$

$$V^{(1)} \equiv \frac{v^{(1)}\delta t}{\delta y} = |V|\,(1-|V|)B - 2\,(1-F)\,U\,V\,A$$

where A and B are numerical estimates at the particular edge. For example, on the right edge of a cell

$$A \cong \left[\frac{\delta x}{2\,\Psi}\frac{\partial\Psi}{\partial x}\right]_{i+1,j+1/2} \qquad (4.7)$$

$$B \cong \left[\frac{\delta y}{2\,\Psi}\frac{\partial\Psi}{\partial y}\right]_{i+1,j+1/2}$$

After finishing a second donor cell pass with the velocities in (4.6) , there will be a smaller residual error. This error can be estimated with a donor cell approximation, leading to a new set of pseudo velocities $U^{(2)}$ and $V^{(2)}$. The process can be repeated continuously, with the pseudo velocities at the k^{th} step being related to those at the $(k-1)^{th}$ step by the recursion relation

$$U^{(k)} = |U^{(k-1)}|\,(1-|U^{(k-1)}|)\,A - 2\,(F)\,U^{(k-1)}\,V^{(k-1)}\,B \qquad (4.8)$$

$$V^{(k)} = |V^{(k-1)}|\,(1-|V^{(k-1)}|)\,B - 2\,(1-F)\,U^{(k-1)}\,V^{(k-1)}\,A$$

with initial conditions

$$U^{(1)} \equiv (|U|-U^2)\,A - 2\,B\,U\,V\,F \qquad (4.9)$$

$$V^{(1)} \equiv (|V|-V^2)\,B - 2\,A\,U\,V\,(1-F)$$

We note that the recursion (4.8) reduces to the one-dimensional case -- see (2.15) -- when B and |V| vanish.

5. TREATING THE CROSS TERM

One of the crucial aspects of solving the recursion relation in one dimension was that the sign of the pseudo velocities did not change. This enables us to eliminate the absolute values in the recursion relation (2.15). In one dimension, maintenance of the sign was a consequence of the Courant condition for stability (3.1) and also the limit on the magnitude of A.

In two dimensions, generalizations of both of these conditions apply. The Courant condition becomes

$$(|U|+|V|) \leq 1 \tag{5.1}$$

Also, for fields that are always positive or always negative,

$$|A| \leq 1 \quad \text{and} \quad |B| \leq 1 \tag{5.2}$$

When the field attains values that are both positive and negative, we can add a positive quantity to them as suggested by Smolarkiewicz and described earlier.

Nevertheless, these conditions are not sufficient to guarantee that the pseudo velocities maintain the same sign for all values of the iteration index. The question we address in this section is -- is there a choice of the parameter F that ensures that

$$\text{sign}\,(U^{(k)}) = \text{sign}\,(A) \quad \text{and} \quad \text{sign}\,(V^{(k)}) = \text{sign}\,(B) \quad \forall\, k \tag{5.3}$$

We assert that this is true. To prove this assertion, we must consider each of four cases separately. These cases are

$$\text{case 1}: A \geq 0 \quad \text{and} \quad B \geq 0 \quad \Rightarrow U^{(k)} \geq 0 \quad \text{and} \quad V^{(k)} \geq 0 \quad \forall\, k$$

$$\text{case 2}: A \geq 0 \quad \text{and} \quad B \leq 0 \quad \Rightarrow U^{(k)} \geq 0 \quad \text{and} \quad V^{(k)} \leq 0 \quad \forall\, k$$

$$\text{case 3}: A \leq 0 \quad \text{and} \quad B \geq 0 \quad \Rightarrow U^{(k)} \leq 0 \quad \text{and} \quad V^{(k)} \geq 0 \quad \forall\, k$$

$$\text{case 4}: A \leq 0 \quad \text{and} \quad B \leq 0 \quad \Rightarrow U^{(k)} \leq 0 \quad \text{and} \quad V^{(k)} \leq 0 \quad \forall\, k$$

In Appendix A, we prove case 1. Furthermore, we show that the two series converge, at least for particular choices of the parameter F, and also that F can always be chosen between 0 and 1. The same method of proof can be applied to the each of the other cases.

6. SUMMING THE RECURSION IN TWO DIMENSIONS

We assume that we have chosen a value of the parameter F such that

$$\text{sign}\,(U^{(k)}) = \text{sign}\,(A) \quad \text{and} \quad \text{sign}\,(V^{(k)}) = \text{sign}\,(B) \quad \forall\, k$$

We wish to evaluate the sums

$$W_u \equiv \sum_{k=1}^{\infty} U^{(k)}$$
$$W_v \equiv \sum_{k=1}^{\infty} V^{(k)} \tag{6.1}$$

As before, we expand each of the pseudo velocities in a polynomial series in two small variables, ξ and η. Because of the cross terms, we must allow products of the two variables to appear in the expansions. For the U-series, we now expand

$$U^{(k)} = \alpha_u^{(k)}\xi + \beta_u^{(k)}\xi^2 + \gamma_u^{(k)}\xi^3 + a_u^{(k)}\xi\eta + b_u^{(k)}\xi^2\eta + c_u^{(k)}\xi\eta^2 + \ldots \tag{6.2}$$

where the Greek coefficients are used when there are no cross terms and the Latin coefficients are used for the cross terms. Similarly for the V series,

$$V^{(k)} = \alpha_v^{(k)}\eta + \beta_v^{(k)}\eta^2 + \gamma_v^{(k)}\eta^3 + a_v^{(k)}\xi\eta + b_v^{(k)}\eta^2\xi + c_v^{(k)}\eta\xi^2 + \ldots \tag{6.3}$$

In each case, we have kept terms up to third order.

It is again simplest to choose the expansion variables to be $U^{(1)}$ and $V^{(1)}$ -- see (4.9). Then the initial conditions take the simple form

$$\alpha_u^{(1)} = 1 \;;\; \beta_u^{(1)} = 0 \;;\; \gamma_u^{(1)} = 0 \;;\; a_u^{(1)} = 0 \;;\; b_u^{(1)} = 0 \;;\ldots.$$
$$\alpha_v^{(1)} = 1 \;;\; \beta_v^{(1)} = 0 \;;\; \gamma_v^{(1)} = 0 \;;\; a_v^{(1)} = 0 \;;\; b_v^{(1)} = 0 \;;\ldots. \tag{6.4}$$

Note that not every possible term has been kept in (6.2) and (6.3). Some terms are excluded by the form of the recursion relation and some are excluded because they have zero initial condition.

To sum, we insert (6.2) and (6.3) into the right hand side of the recursion relations (4.8). Since $U^{(k+1)}$ and $V^{(k+1)}$ both have series similar to (6.2) and (6.3), we derive the new recursion relations by matching coefficients of the various powers of ξ and η. From the U-recursion, we find for that the terms without η satisfy the same recursion relations as the one-dimensional case -- see (3.6)

$$\begin{aligned} \alpha_u^{(k+1)} &= |A|\, \alpha_u^{(k)} \\ \beta_u^{(k+1)} &= \{ |A|\, \beta_u^{(k)} - A\, \alpha_u^{(k)} \alpha_u^{(k)} \} \\ \gamma_u^{(k+1)} &= \{ |A|\, \gamma_u^{(k)} - 2\, A\, \alpha_u^{(k)} \beta_u^{(k)} \} \end{aligned} \tag{6.5}$$

while the coefficients of the cross terms obey the set

$$\begin{aligned} a_u^{(k+1)} &= |A|\, a_u^{(k)} - 2\, B\, F\, \alpha_u^{(k)}\, \alpha_u^{(k)} \\ b_u^{(k+1)} &= \{ |A|\, b_u^{(k)} - 2\, A\, \alpha_u^{(k)}\, a_u^{(k)} - 2\, B\, F\, a_u^{(k)}\, \beta_u^{(k)} \} \\ c_u^{(k+1)} &= \{ |A|\, c_u^{(k)} - 2\, B\, F\, \alpha_u^{(k)}\, b_u^{(k)} \} \end{aligned} \tag{6.6}$$

The recursion relations (6.5) are the same as (3.6), with the same initial conditions. Thus the sums associated with the α-series, β-series, etc., are identical with those the one-dimensional results -- c.f. (3.15) -- except for the definition of the expansion variable ξ which is now given by (4.6). The sums associated with the cross terms can be evaluated by the same techniques. Our final result for the recursive velocity W_u is

$$\begin{aligned} W_u = {} & [\Sigma_\alpha]\, \xi + [\Sigma_\beta]\, \xi^2 + [\Sigma_\gamma]\, \xi^3 \\ & + [\Sigma_a]\, \xi\eta + [\Sigma_b]\, \xi^2\eta + [\Sigma_c]\, \eta^2\xi + \ldots \end{aligned} \tag{6.7}$$

where

$$\begin{aligned} \Sigma_\alpha &= \frac{1}{(1 - |A|)} \\ \Sigma_\beta &= \frac{-A}{(1 - |A|)(1 - A^2)} \\ \Sigma_\gamma &= \frac{2\, |A|^3}{(1 - |A|)(1 - A^2)(1 - |A|^3)} \end{aligned} \tag{6.8}$$

and the cross term coefficients

$$\Sigma_a = \frac{-2\,B\,F}{(1-|A|)(1-|AB|)}$$

$$\Sigma_b = \frac{2\,A\,B\,F}{(1-|A|)(1-A^2|B|)}\left[\frac{|B|}{(1-|AB|)} + \frac{2\,A}{(1-A^2)}\right] \tag{6.9}$$

$$\Sigma_c = \frac{2\,|A|\,B^2\,F}{(1-|A|)(1-|A|B^2)(1-|AB|)}$$

The final results for the pseudo velocity W_v can be found from (6.7), (6.8), and (6.9) by systematically interchanging

$$\xi \leftrightarrow \eta \quad ; \quad A \leftrightarrow B \quad , \quad F \leftrightarrow 1-F \tag{6.10}$$

We close this section by noting that our analysis has imposed a fairly narrow range of allowable values of the parameter F. However these requirements are derived from our method of analysis. In Sec. 7.2 we discuss this subject further.

7. IMPLEMENTATION

In this section we address subjects associated with the implementation of the recursion sums. First there is the question of limiting the magnitude of the pseudo velocities. This point will be discussed in Sec. 7.1. We shall also return briefly to the question of how to choose the constant F, defined in (4.5), which distributes the cross term between the two components of the antidiffusive velocity.

7.1 Limiting the Recursive Velocities

We begin by noting that MPDATA depends on the positive definiteness of the donor cell method. Furthermore, the positive definite property depends on suitably limiting the velocities. When the advection velocity is constant in space, then a sufficient condition for positive definiteness [1] is that the Courant number is bounded by unity -- i.e.,

$$C \equiv \left|\frac{u\,\delta t}{\delta x}\right| \leq 1 \qquad (7.1)$$

When the velocity varies as a function of coordinate, it is necessary to add a second condition relating to the divergence of the velocity field

$$D \equiv \left|\nabla \circ u\,\delta t\right| \leq 1 \qquad (7.2)$$

In a simple one pass donor cell scheme, both these conditions can be satisfied in every cell by a suitable choice of the time step δt .

The formalism discussed in this report results in the two-pass scheme

$$\begin{aligned} \Psi^{(1)}_{i+1/2} &= \Psi^{n}_{i+1/2} - [\,F(\Psi^{n}_{i+1/2}, \Psi^{n}_{i+3/2}, U_{i+1}) - F(\Psi^{n}_{i-1/2}, \Psi^{n}_{i+1/2}, U_i)\,] \\ \Psi^{n+1}_{i+1/2} &= \Psi^{(1)}_{i+1/2} - [\,F(\Psi^{(1)}_{i+1/2}, \Psi^{(1)}_{i+3/2}, W^{(1)}_{i+1}) - F(\Psi^{(1)}_{i-1/2}, \Psi^{(1)}_{i+1/2}, W^{(1)}_{i})\,] \end{aligned} \qquad (7.3)$$

with sufficient conditions to insure stability and positive definiteness of each pass separately -- that is,

$$\begin{aligned} |U| &\leq 1 \quad \text{and} \quad \left|\nabla \circ u\,\delta t\right| \leq 1 \\ |W| &\leq 1 \quad \text{and} \quad \left|\nabla \circ w\,\delta t\right| \leq 1 \end{aligned} \qquad (7.4)$$

We note that, to second-order accuracy, we could use a recursive velocity W ^{n} based on the n-time level field Ψ^n. Such a scheme could be written as a single-pass algorithm

$$\begin{gathered}\Psi^{n+1}_{i+1/2} = \Psi^{n}_{i+1/2} \\ [F(\Psi^{n}_{i+1/2}, \Psi^{n}_{i+3/2}, U_{i+1} + W^{n}_{i+1}) - F(\Psi^{n}_{i-1/2}, \Psi^{n}_{i+1/2}, U_{i} + W^{n}_{i})]\end{gathered} \tag{7.5}$$

Sufficient conditions for stability and positive definiteness of (7.5) would be

$$|U + W| \le 1 \quad \text{and} \quad \left|\nabla \circ (u + w)\,\delta t\right| \le 1 \tag{7.6}$$

One concern in using the two-pass method (7.3) instead of the single- pass method (7.5) is that it may be computationally more expensive In practice we have found the extra expense to be negligible. On the other hand, there are several important advantages to the two-pass method.

The biggest advantage of (7.3) is realized when the flow has steep gradients. In this case, arguments about order of accuracy become unimportant since the quantities upon which the Taylor series expansions are based may not be small. In this regime, (7.3) allows a larger flux than (7.5) and better preserves regions of steep gradients.

A second advantage lies in preservation of symmetry. (7.6) is not symmetric in its limiting of the velocity. For example, in a triangular wave being transported to the right, the antidiffusive velocity on the leading edge is negative (i.e., the same sign as the slope) whereas on the trailing edge it is positive. After limiting, the leading edge oversteepens and the trailing edge is rounded and the symmetry of the original triangle is lost. This problem is much less severe when scheme (7.3) is used, mainly because the limiting in (7.4) is more symmetric.

Smolarkiewicz uses (7.3) in his MPDATA [1] Since the antidiffusive velocity in his scheme is given by

$$W = U(1 - U)\frac{\Psi^{(1)}_{i+1/2} - \Psi^{(1)}_{i-1/2}}{\Psi^{(1)}_{i+1/2} + \Psi^{(1)}_{i-1/2}} \tag{7.7}$$

whenever the field is positive and U is properly bounded, then |W| < 0.25. This is sufficient to guarantee positive definiteness of the second pass and no additional limiting on W is required. We also note that in this scheme, the antidiffusive velocity of successive passes (i.e., a third or fourth corrective iteration) is bounded by the antidiffusive velocity of the previous pass, and so positive-definiteness is guaranteed for an arbitrary number of iterations.

After the recursive velocities are approximately summed, the new pseudo velocities are not necessarily properly bounded. For example in (3.15) each term contains the factor (1 - |A|) in the denominator. Since |A| may be arbitrarily close to 1, these terms are actually unbounded. In practice then, after calculating the new pseudo velocity and before calculating the fluxes, it is necessary to limit them. This is accomplished with a FORTRAN statement like

$$W = \text{amax1}\,[\,-0.5,\ \text{amin1}\,[0.5, W]\,] \qquad (7.8)$$

Both amax1 and amin1 are vectorizable functions and are very efficient.

The situation in two dimensions is presently less satisfactory. The analogue of (7.1) is

$$\left|\frac{u\,\delta t}{\delta x}\right| + \left|\frac{v\,\delta t}{\delta y}\right| \leq 1 \qquad (7.9)$$

where (u,v) are the two components of the velocity vector. In addition, restriction based on divergence must still apply. A simple solution would be to restrict

$$\begin{aligned} W_u &= \text{amax1}\,[\,-0.25,\ \text{amin1}\,[0.25, W_u]\,] \\ W_v &= \text{amax1}\,[\,-0.25,\ \text{amin1}\,[0.25, W_v]\,] \end{aligned} \qquad (7.10)$$

for the dimensionless antidiffusive velocity (W_u, W_v). This is sufficient to guarantee positive definiteness, but is not necessary and in practice we find it is too restrictive in the sense of unnecessarily diffusing the solution.

By analogy with the original MPDATA, we restricted the antidiffusive velocities by the physical velocities --

$$\begin{aligned} W_u &= \text{amin1}\,[\,|U|, |W_u|\,] * \text{sign}(1., W_u) \\ W_v &= \text{amin1}\,[\,|V|, |W_v|\,] * \text{sign}(1., W_v) \end{aligned} \qquad (7.11)$$

We have verified experimentally that this choice works and is less restrictive than (7.10).

It is possible to formulate even less restrictive limits, and these should yield better results. However such limits, as regards the divergence, are nonlocal -- i.e., will involve four velocities at a time -- and may cost more in computer time to implement. This point is still under investigation.

7.2 Distributing the Cross Term

The parameter F that appears in our results (6.9) for the recursive velocities in two dimensions represents the portion of the truncation cross term that is compensated in the u-component of the pseudo velocity. Our analysis in Sec. 6 required that F be chosen from a fairly limited range. However that requirement is based on our method of summing the series, and in particular upon our treatment of the absolute value signs that appear in the recursion.

On the other hand, the final results (6.9) appear to be well-behaved for a much wider range of the choice of F. Theoretically we speculate that our results can be analytically continued in the parameter F beyond the range for which they are derived. In a practical sense, we have verified numerically that the implementation of the recursive velocities is insensitive to the choice of F (at least in the range $0 < F < 1$), and we have chosen to use $F = 0.5$ in all cases. Smolarkiewicz makes the same choice in [1] in his iterative scheme.

8. NUMERICAL RESULTS

We have implemented our new formulae for the antidiffusive velocities in two codes. The first is a "convergence" code in which a simple Gaussian pulse is advected in one dimension. In this code we are able to estimate both a global truncation error and rate of change of the error as the spatial resolution and the Courant number are changed.

The second is a two-dimensional advection code for solid body rotation. Our prototype problem here is six full rotations of a cone, requiring 3768 time steps. We show results here using our two-dimensional recursive velocities in a fully two-dimensional calculation. We have also solved the two-dimensional problem using our one-dimensional formulae in a time-split code. The results of the time split calculations are comparable to the two-dimensional calculations and are not shown among our examples.

All of our calculations use the two-pass scheme (7.3) in which the recursive velocities depend on the field values $\Psi^{(1)}$ -- i.e., the field values after the first donor cell pass.

8.1 One-Dimensional Convergence Code

Smolarkiewicz' convergence test [3] is a powerful tool for computing the global convergence rate of a numerical advection scheme. The code calculates the advection of a simple Gaussian pulse with constant velocity in one dimension. The exact solution of this problem is simply the translation of the pulse in space with no change in shape. Thus the code can compare the numerical solution with the exact solution and construct a single global estimate of the truncation error in an integral norm.

Within the code, the same problem is run varying both the total number of grid points and the Courant number. In particular, the size of the grid is varied from

$$\Delta x \in \left(\frac{\Delta x_8}{2^0}, \frac{\Delta x_8}{2^1}, \frac{\Delta x_8}{2^2}, \frac{\Delta x_8}{2^3}, \frac{\Delta x_8}{2^4}, \frac{\Delta x_8}{2^5}, \frac{\Delta x_8}{2^6}, \frac{\Delta x_8}{2^7}\right) \qquad (8.1)$$

Also the Courant number is varied from

$$C \in (0.05, 0.10, 0.15, 0.20, \ldots. 0.90, 0.95) \qquad (8.2)$$

From each of these 152 runs, we construct a function T(C,Δx) that is the logarithm base 2 of the global truncation error.

Lines of constant value -- i.e., isolines -- of this function T(C,Δx) are plotted on a polar graph in which the radius and angle are defined by

$$r = \ln_2 [\frac{\Delta x}{\Delta x_8}] + 8$$
$$\phi = C \frac{\pi}{2} \qquad (8.3)$$

There is much information to be gained from each graph and by comparing graphs. The actual value of the function T(C,Δx) is a measure of the overall accuracy of the calculation. When comparing the graphs of two different algorithms, the algorithm with the more negative value of T is the more accurate. (Such a comparison must be made at a particular point, which means for a particular value of C and Δx.)

Along radial lines -- i.e., lines of constant ϕ -- the resolution is changing while the Courant number remains fixed. The slope dT/dr for constant ϕ represents the convergence rate of the scheme. When we say a scheme has second-order convergence, we mean that $dT/dr \approx 2$.

It is also interesting to note the shape of the isolines. If these lines are isotropic, then the truncation error is independent of polar angle and thus of Courant number. In this case we say the convergence is uniform -- i.e., does not depend on how Δx and Δt go to zero. Donor cell schemes are more accurate for larger Courant numbers and so the surfaces are skewed.

8.2 One-Dimensional Calculations

Figures 8.1 through 8.4 were generated using the convergence code described above. The first three figures are results from the standard version of MPDATA, and the fourth is MPDATA using the new recursive velocities. In particular, Fig. 8.1 is MPDATA with two iterations -- i.e., regular donor cell plus one corrective iteration. Figure 8.2 is MPDATA with three iterations and Fig. 8.3 is MPDATA with four iterations. Comparing these graphs we see that using the new pseudo velocities is roughly equivalent to using the standard MPDATA with four iterations. Also, further increases of iteration number has very little effect in the standard MPDATA.

An interesting feature of Figs. 8.3 and 8.4 is the groove formed by the isolines around $\phi = 45^{o}$. When the second-order truncation error is reduced sufficiently, the third-order error becomes dominant. The third-order error can be shown to be proportional to a polynomial in Courant number

$$\text{third order error} \approx (1 - 3 C + 2 C^2) \qquad (8.4)$$

This polynomial has a zero for C = 0.5, which is $\phi = 45^o$ on the plot, and so the third-order error vanishes here. This results in the groove.

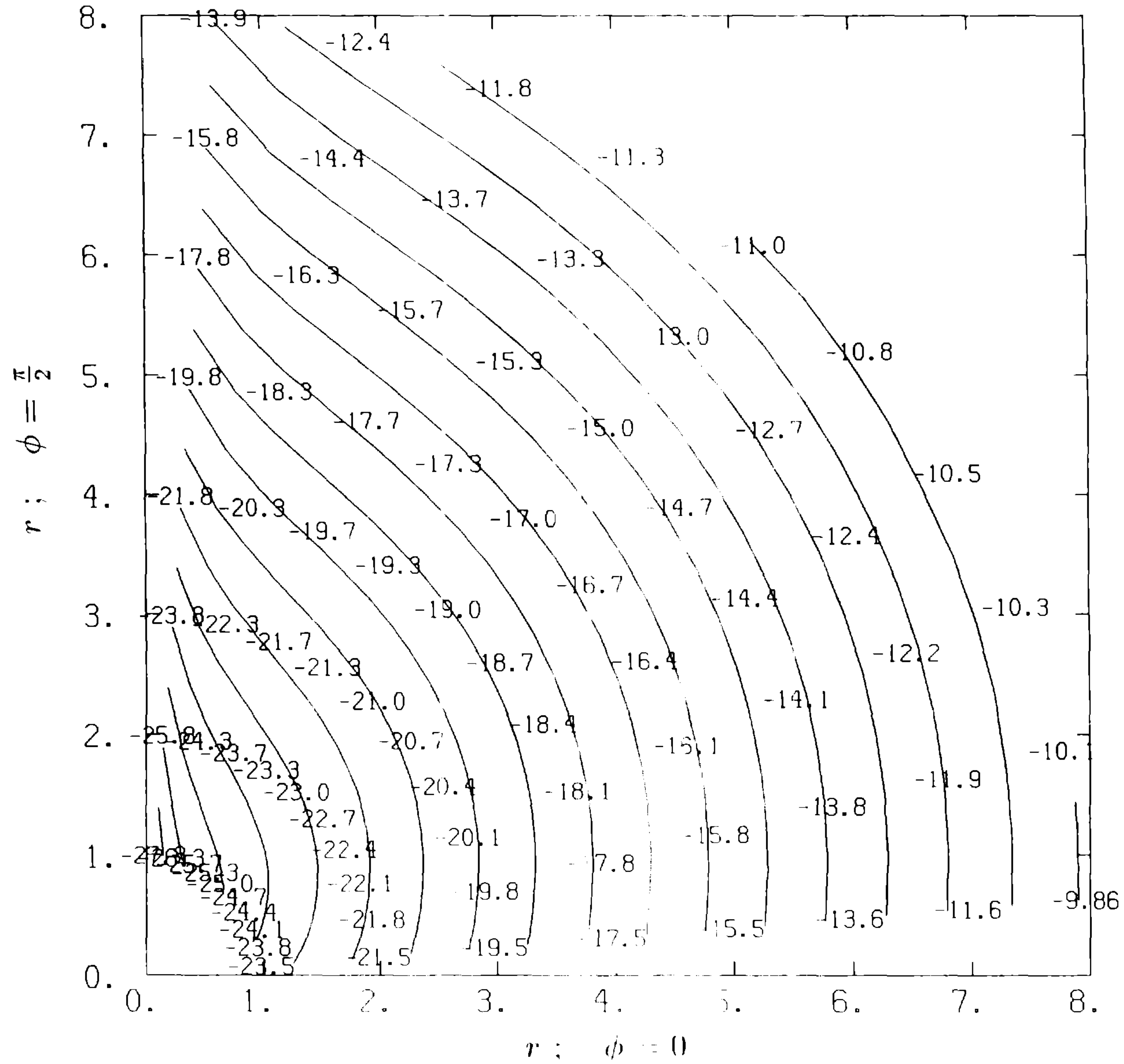

Figure 8.1: Isolines for the convergence test for transport of a Gaussian pulse using standard MPDATA with two iterations.

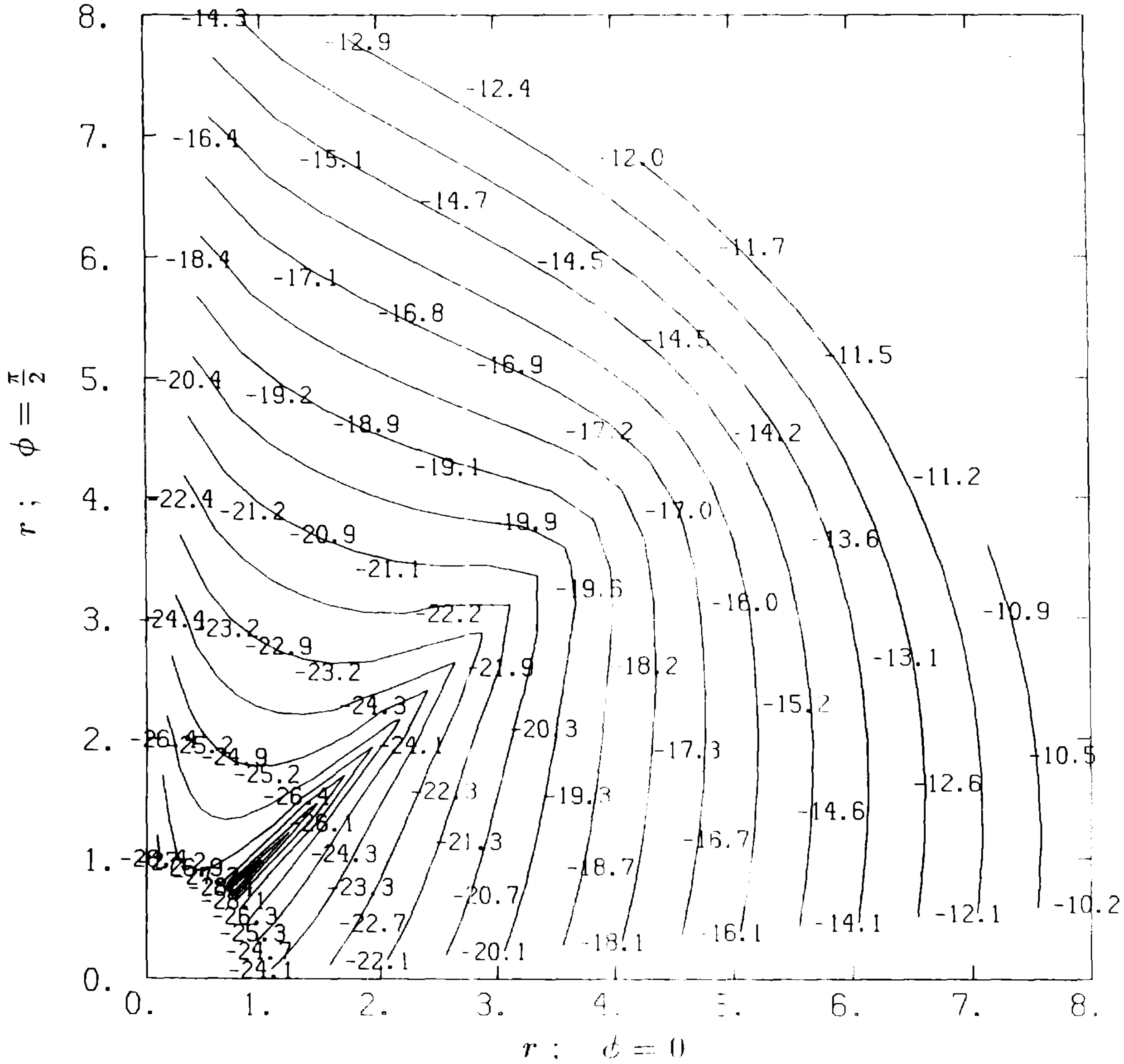

Figure 8.2: Isolines for the convergence test for transport of a Gaussian pulse using standard MPDATA with three iterations.

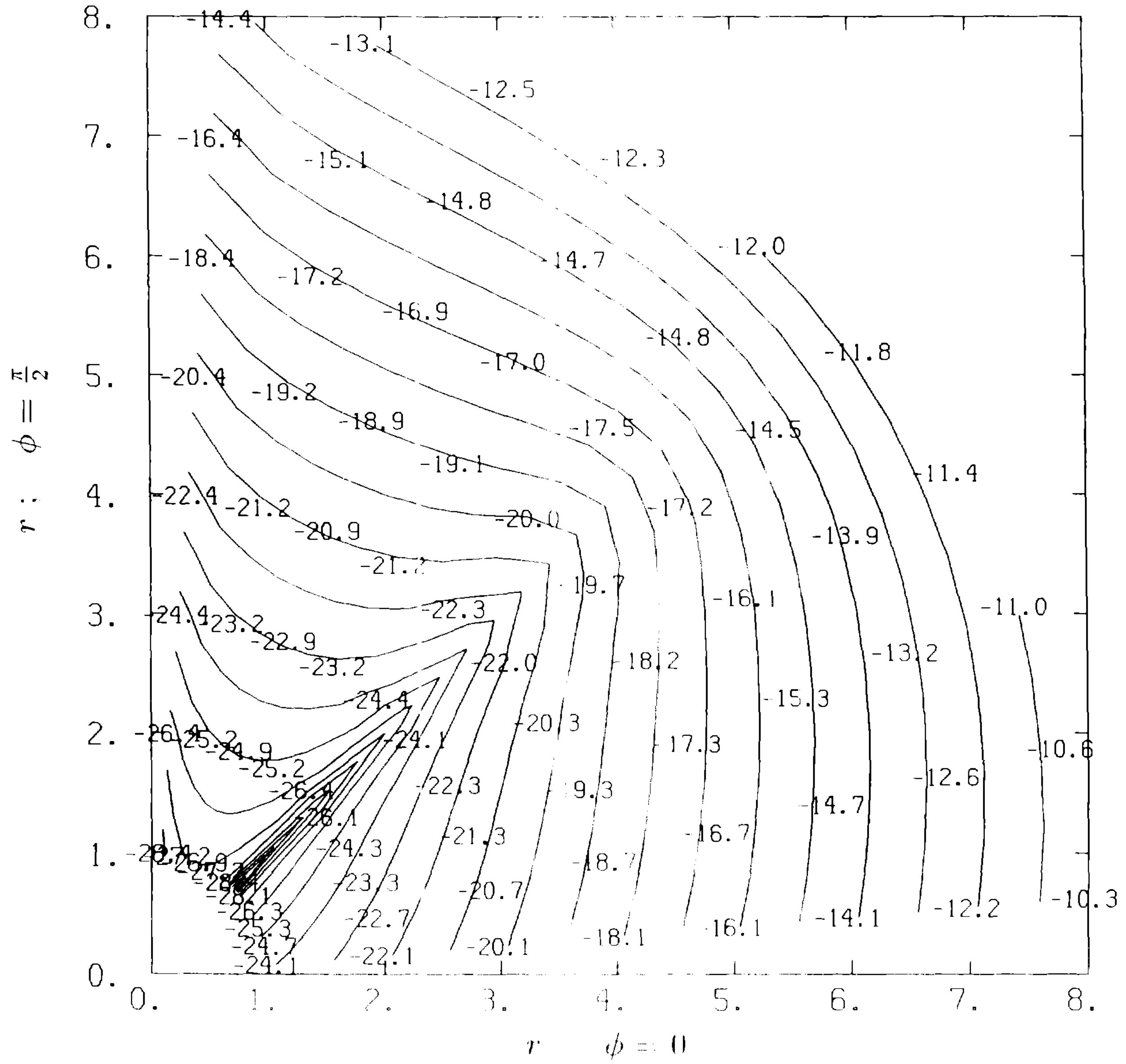

Figure 8.3: Isolines for the convergence test for the transport of a Gaussian pulse using standard MPDATA with four iterations.

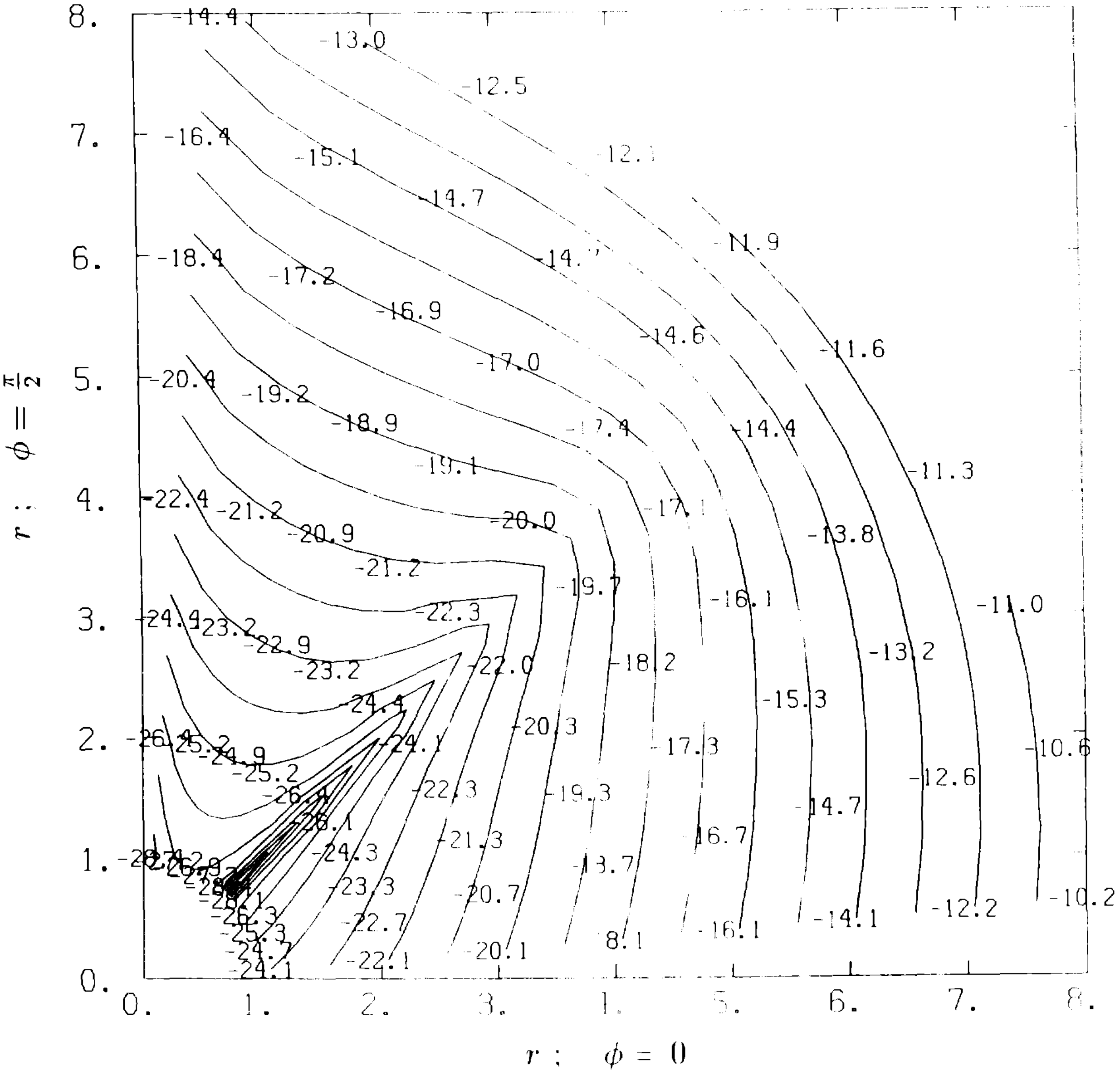

Figure 8.4: Isolines for the convergence test for the transport of a Gaussian pulse using MPDATA with the recursive pseudo velocities. Note that this plot is roughly equivalent to MPDATA with four iterations.

To further illustrate this point, we added a donor cell approximation to the third-order error in the convergence code. This term in the pseudo velocity is proportional to the second spatial derivative of the field. It is not necessary to include this term in the recursion relations to achieve a third-order convergence rate. If one included this term, it would further reduce the magnitude of the error, but we have not done so.

The results are shown in Figs. 8.5, 8.6, and 8.7. Figure 8.5 is standard MPDATA with two iterations and a correction for the third-order error. This is not sufficient to effectively cancel all the second-order error which continues to dominate. Thus the calculation shows a second-order convergence. Figure 8.6 is similar with three iterations. Here the second-order error has been mostly eliminated and the calculation shows a third-order convergence. Figure 8.7 is MPDATA with the third-order corrections and our new recursive velocities. It also shows a third-order convergence.

8.3 Two-Dimensional Test Problems

The two-dimensional rotation test uses a square mesh of 101 by 101 points. Both $\Delta x = \Delta y = 1.0$, whereas $\Delta t = 0.1$. The angular velocity $\omega = 0.1$ and the velocity components are

$$(u,v) = -\omega(y - y_0, x_0 - x) \qquad (8.5)$$

Here the center of rotation (x_0, y_0) is the center of the mesh ($50\,\Delta x$, $50\,\Delta y$). The maximum Courant number (5.1) is 0.99 and one full rotation requires 628 time steps. The initial cone is centered at the point ($75\,\Delta x$, $50\,\Delta y$). The cone has an initial base diameter of 15 and a height of 4. Solutions to all the problems are shown after six full rotations.

Although the two-dimensional solutions contain much detail, we have chosen to make comparisons only on two features of the solutions. In a quantitative sense, we compare the maximum values of the field between different calculations. Qualitatively, we compare the front to back symmetry of the cone.

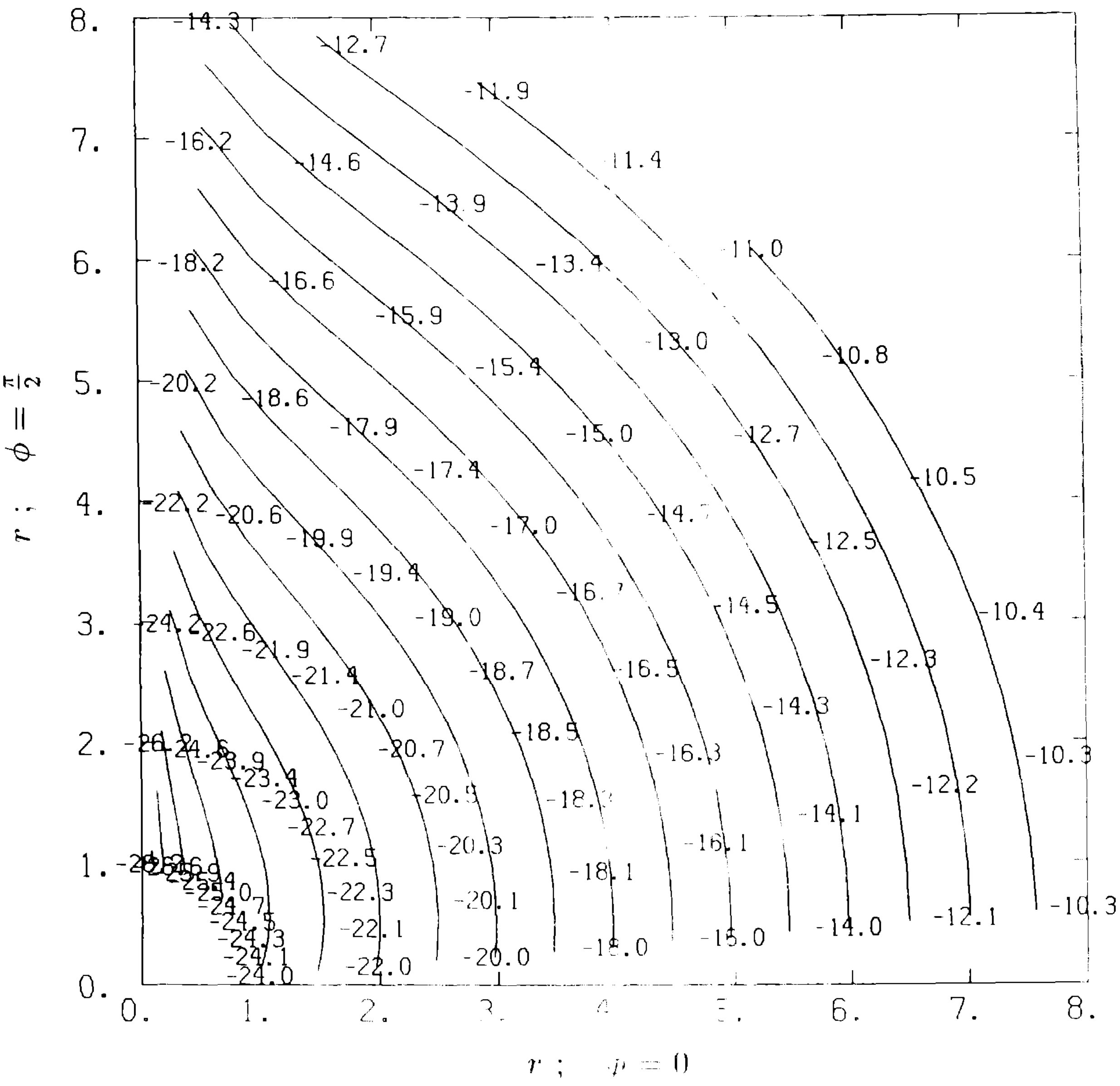

Figure 8.5: Isolines for the convergence test for the transport of a Gaussian pulse using MPDATA with third-order corrections and with two iterations.

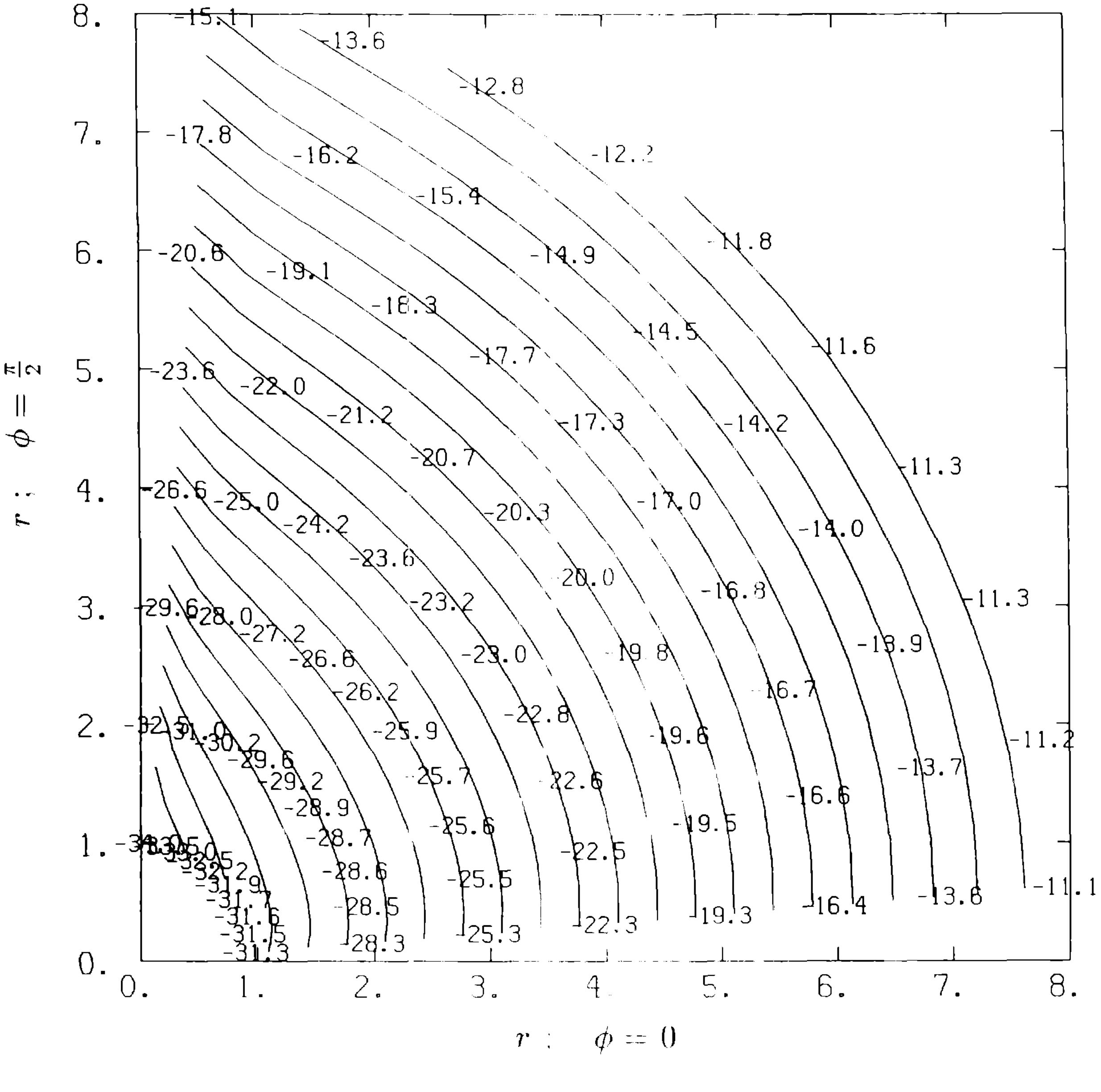

Figure 8.6: Isolines for the convergence test for the transport of a Gaussian pulse using MPDATA with third-order corrections and with three iterations.

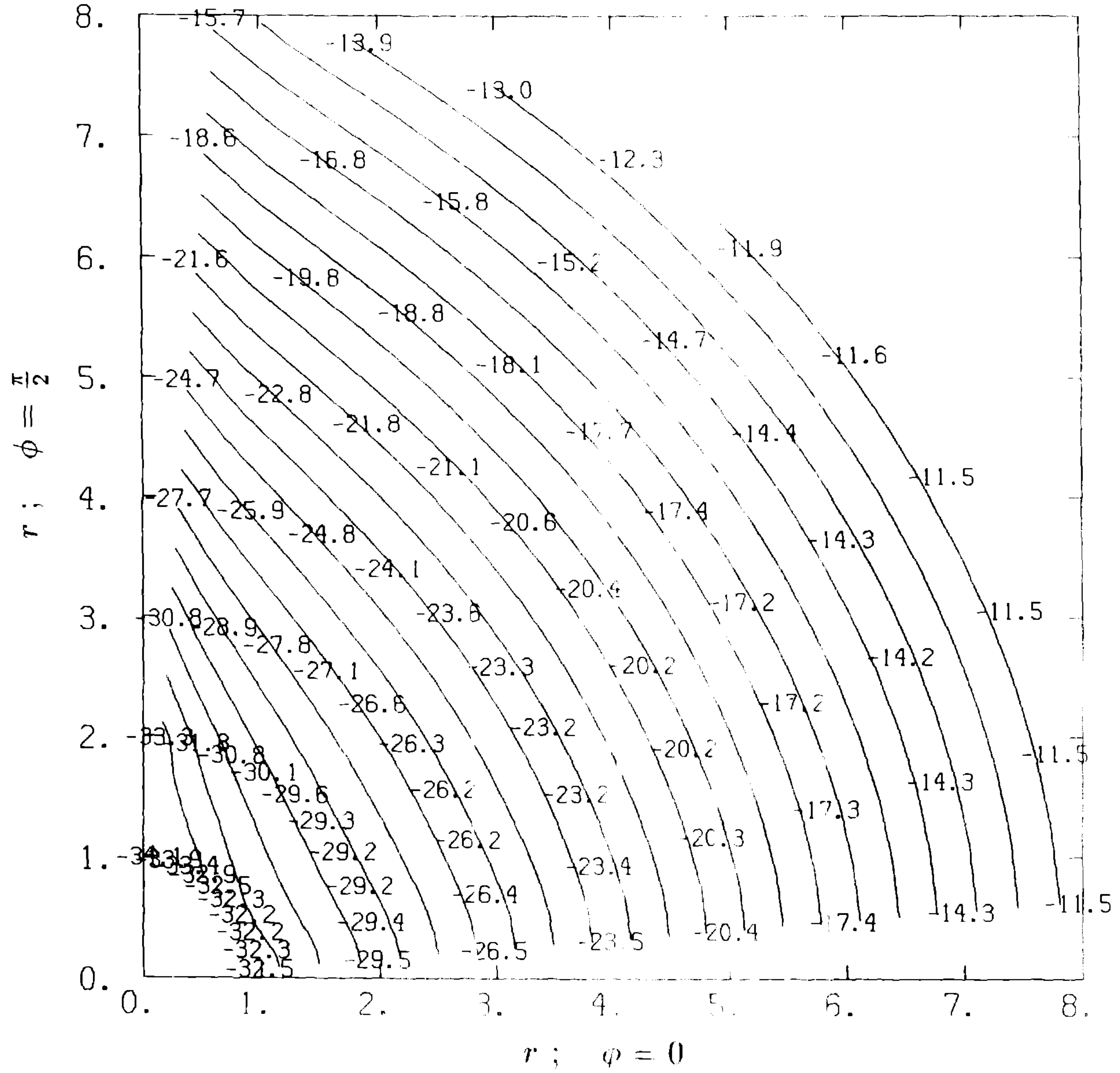

Figure 8.7: Isolines for the convergence test for the transport of a Gaussian pulse using MPDATA with third-order corrections and with the new recursive pseudo velocities.

8.4 Two-Dimensional Test Results

In Fig. 8.8 we show a standard MPDATA run with two iterations; in Fig. 8.9 we show a standard MPDATA run with three iterations; in Fig. 8.10 we show a standard MPDATA run with four iterations; in Fig. 8.11 we show MPDATA with the new recursive velocities. Qualitatively from the figures and more quantitatively from Table 8.1, we see that using the new pseudo velocities matches most closely to using four iterations.

TABLE 8.1

Run #	Iterations	Recursive	Maximum
1	2	no	2.179
2	3	no	3.156
3	4	no	3.261
4	2	yes	3.263

Table 8.1 -- compares the maximum value of a cone after one full revolution of a cone. The initial maximum value is 4.0. The use of the recursive velocities maintains a maximum value equivalent to the original MPDATA scheme with a total of four iterations.

Figures 8.12 and 8.13 are the same calculations as described in the previous paragraph except that the third-order corrections are implemented. In Fig. 8.12 we show the original MPDATA with three iterations. In Fig. 8.13 we show the calculation using the new recursive velocities. The most noticeable feature is that the symmetry of the cone is considerably improved.

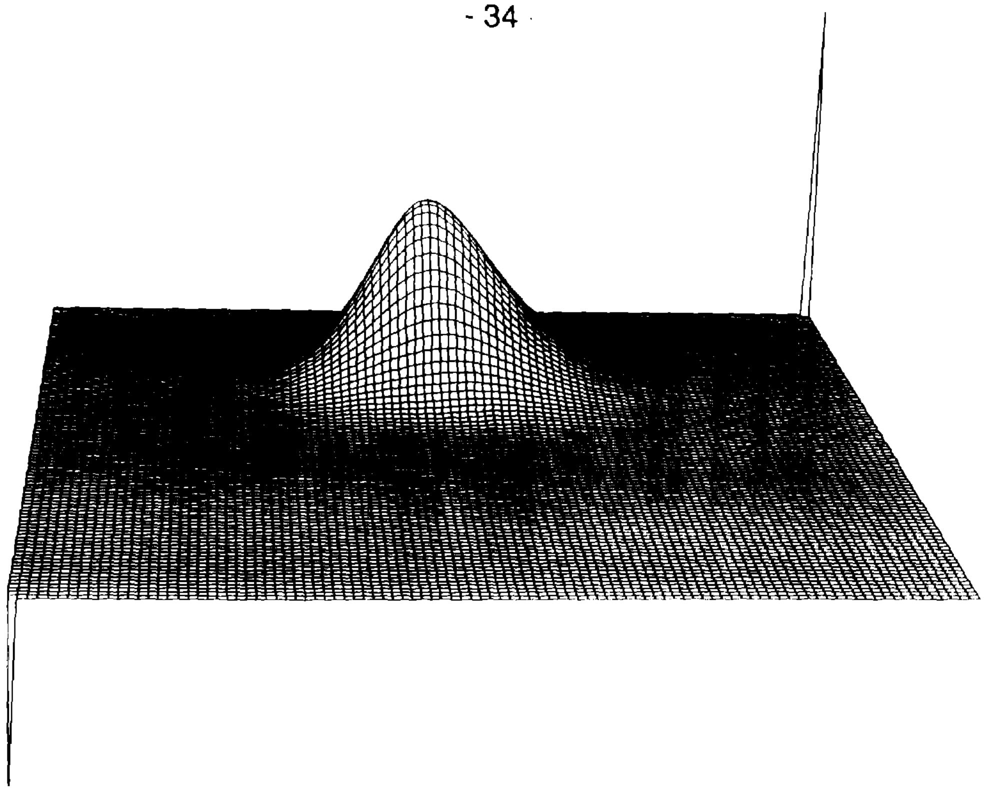

Figure 8.8 shows a cone that has been advected through six full rotations and 3768 timesteps using a standard two-dimensional MPDATA scheme with two iterations. The original height of the cone was 4.0.

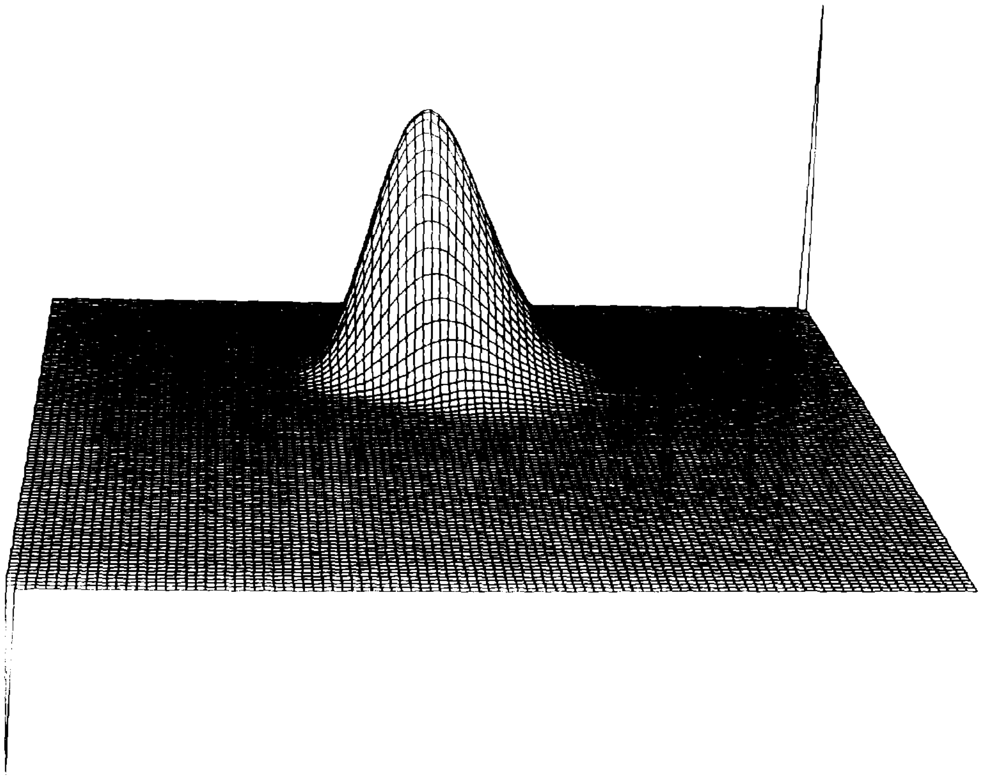

Figure 8.9 shows a cone that has been advected through six full rotations using a standard two-dimensional MPDATA scheme with three iterations.

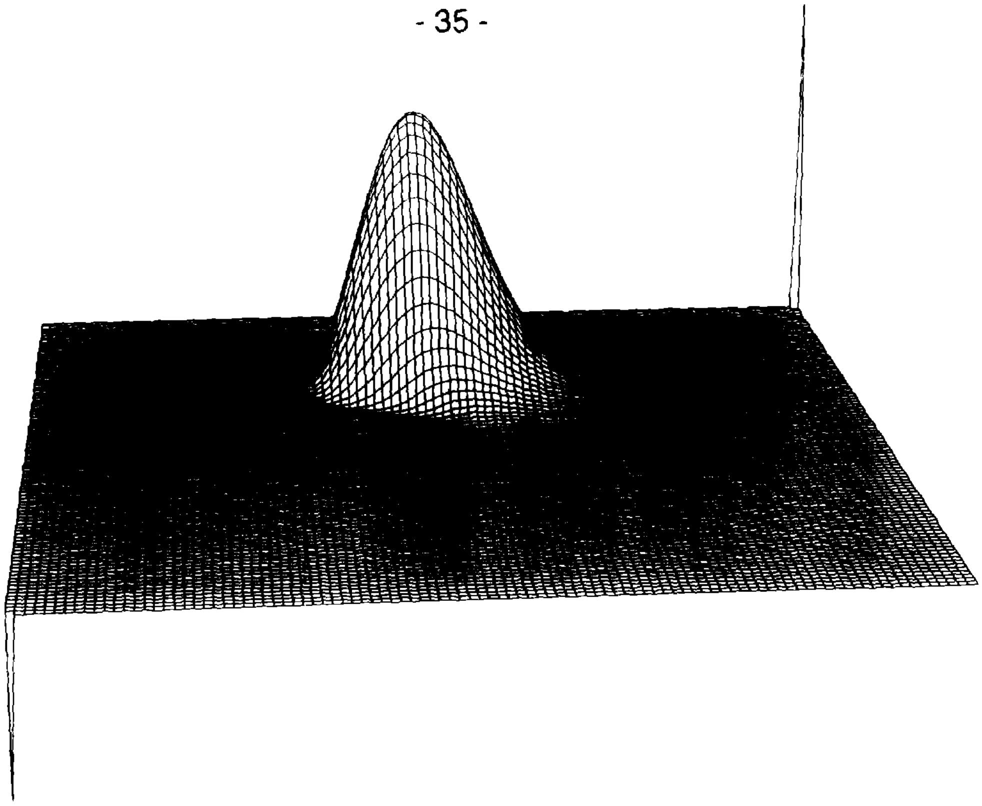

Figure 8.10 shows a cone that has been advected through six full rotations using a standard two-dimensional MPDATA scheme with four iterations.

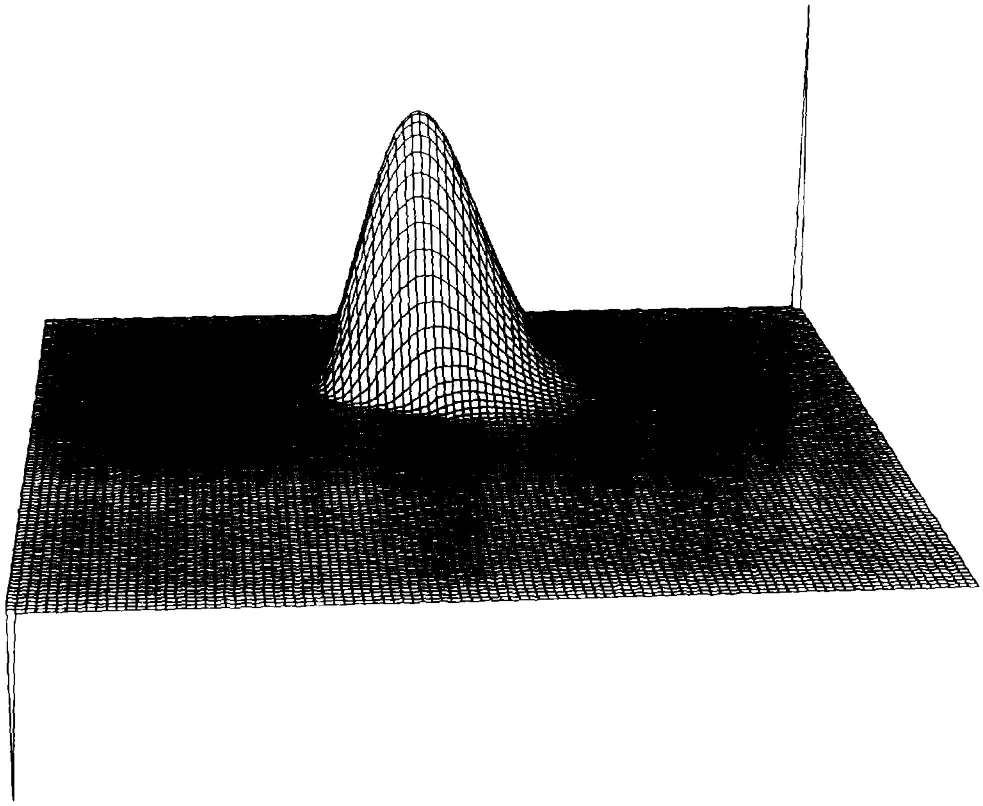

Figure 8.11 shows a cone that has been advected through six full rotations using MPDATA with the two-dimensional recursive pseudo velocities.

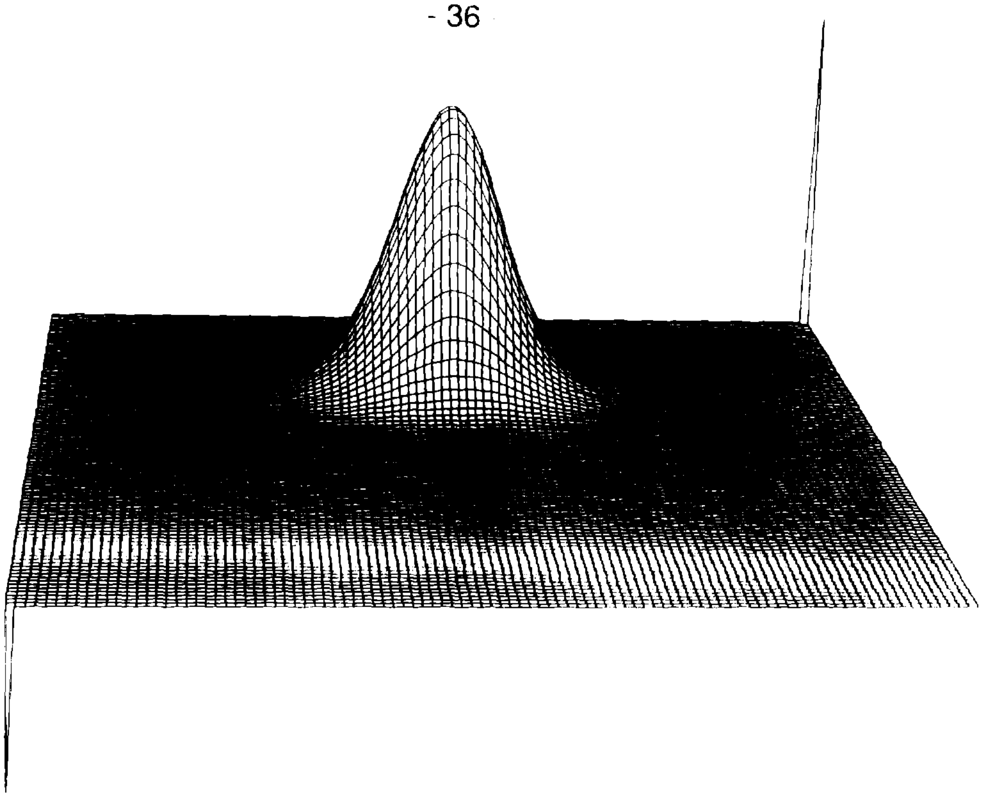

Figure 8.12 shows a cone that has been advected through six full rotations using a two-dimensional MPDATA scheme with third-order corrections and three iterations.

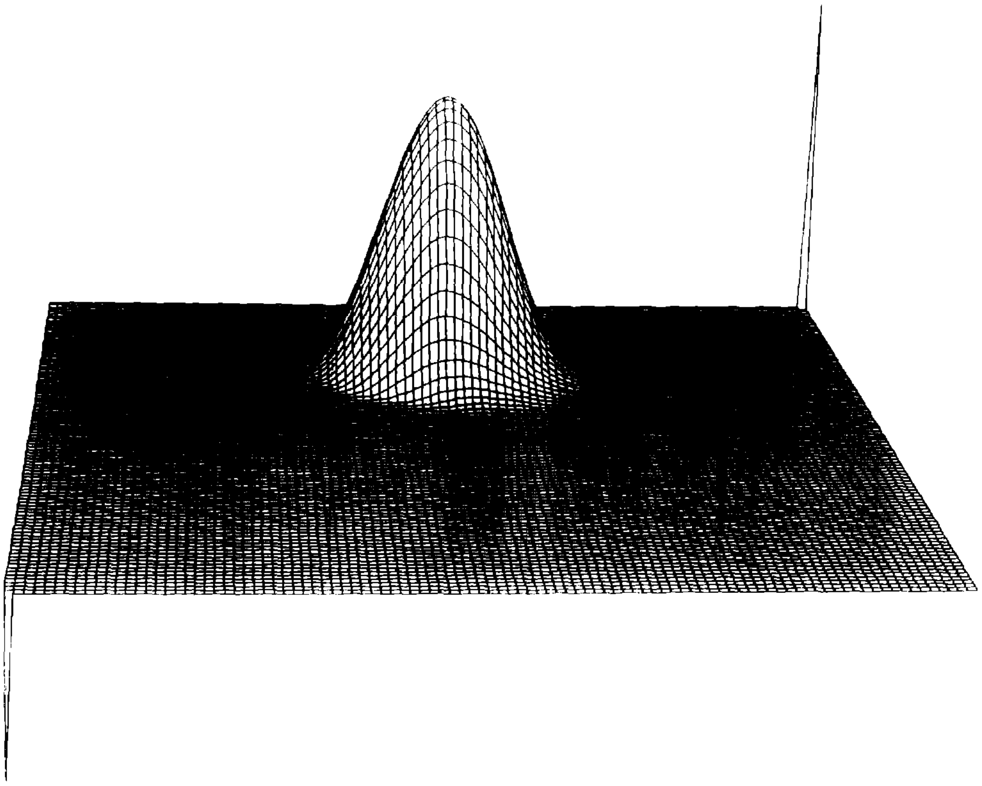

Figure 8.13 shows a cone that has been advected through six full rotations using MPDATA with third-order corrections and the two-dimensional recursive pseudo velocities.

8.5 Comparing the Efficiency of Iterations and Recursive Velocities

In this section we want to use the CPU (execution time of the central processing unit measured in seconds) for various problems as a measure for comparing the overall efficiency of the standard MPDATA with iterations to MPDATA using the recursive velocities. Our calculations were run on a CRAY XMP. Essentially all the calculations were vectorized. We mention three caveats to keep in mind.

1) The results of our comparisons must reflect the details of our implementations. Although we spent some time attempting to optimize code, we believe that further optimization is possible, both for the calculations with iterations and for the calculations with the recursive velocities.
2) The complexity of the recursive velocities plays a major role in the comparisons. This complexity is influenced by the number of terms that one chooses to keep in the expansions. In turn, the number of terms depends both on the accuracy one wishes to realize and on the Courant number of the calculation. By our choice of the expansion parameters ξ and η, the series will converge most rapidly for Courant numbers near 0. and near 1., and will converge most slowly for Courant numbers near 0.5.
3) The CPU numbers are reproducible only to within about plus or minus 5 %. This is due to factors concerning the exact state of the CRAY during execution time.

We compare eight problems. We have chosen to use the two-dimensional solid body rotations. The first three runs are MPDATA with two, three, and four iterations (Figs. 8.8, 8.9, and 8.10). The fourth run will be MPDATA using the recursive velocities (Fig. 8.11). The fifth run is the original MPDATA scheme with three iterations and the third-order corrections turned on (Fig. 8.12). The sixth run uses the new recursive velocities with third-order corrections (Fig. 8.13) The seventh run is the original MPDATA scheme with three iterations, third-order corrections and a nonoscillatory option turned on. The eighth run uses the new recursive velocities with third-order corrections and a nonoscillatory option. We have not shown the solutions for the cone when the nonoscillatory option is turned on since for this case, the original MPDATA is already nonoscillatory and the comparable results are hardly distinguishable. However, for other initial conditions -- i.e., like a cylinder -- one can see significant overshoots unless the nonoscillatory option is turned on

In Table 8.2, four runs with standard MPDATA are compared. We see that the extra complexity of the recursive velocities costs about the same as one extra iteration. We have already noted that for equivalent accuracy, two extra iterations would be required and so the recursive velocities are slightly more efficient.

TABLE 8.2

run	iterations	recursive	order	FCT	CPU (sec)
1	2	no	2	no	37.0
2	3	no	2	no	64.2
3	4	no	2	no	91.3
4	2	yes	2	no	67.2

Table 8.2 -- compares the CPU on a CRAY computer. All the runs here are the original MPDATA with varying numbers of iterations except the last run which uses the new recursive velocities. The running time using the new recursive velocities is comparable to the original MPDATA with three total iterations.

The edge becomes more clear when various options are added to MPDATA. Essentially we are increasing the total amount of work that is done during each iteration, and decreasing the relative cost of the recursive velocities. In Table 8.3 we compare the two runs when third-order corrections are included. The original MPDATA is 30% more expensive when three iterations are used. If four iterations were used, the savings would be even greater

TABLE 8.3

run	iterations	recursive	order	FCT	CPU (sec)
5	3	no	3	no	130.3
6	2	yes	3	no	101.7

Table 8.3 -- compares the CPU on a CRAY computer. In these runs we turn on the third-order corrections, comparing the new recursion velocities with the original iteration scheme using three iterations. Due to the extra work in computing the third-order corrections, the calculation using the new recursive velocities is significantly faster than the original MPDATA with three iterations. The comparison would look even better if we used original MPDATA with four iterations.

In Table 8.4 we compare the two runs when the nonoscillatory option FCT is included as well as the third-order corrections. FCT is itself fairly expensive and the savings using the recursive velocities is approaching 50%.

TABLE 8.4

run	iterations	recursive	order	FCT	CPU (sec)
7	3	no	3	yes	210.3
8	2	yes	3	yes	145.0

Table 8.4 -- compares the CPU on a CRAY computer. In these runs we compare calculations using both third-order corrections and a nonoscillatory option. Using the recursive velocities results in a substantial over using the original MPDATA with three iterations.

9. Conclusions

In this paper we describe a major enhancement to the MPDATA [1,2,3] family of advection schemes. Our principal result is a new expression for the antidiffusive velocity. We describe a two-pass algorithm that has accuracy equivalent to the original MPDATA using four or more iterations. Although the form of the new antidiffusive velocity is more complex than the original, we still can demonstrate some small savings of computer time for the simplest MPDATA calculations. When the original MPDATA algorithm is enhanced, with third-order corrections or a nonoscillatory option, the computer time savings become more substantial -- i.e., up to 50%.

The new antidiffusive velocities are based on an analytic estimation of summing the individual velocities from many iterations. The antidiffusive velocities satisfy a nonlinear recursion relation, and the presence of absolute values within the recursion introduce some subtle complications into the analysis. Our analysis is based on a perturbation series approach, and convergence of the series depends upon suitably bounding certain terms. In general, we believe our analysis contributes to a better understanding of the MPDATA methods

On the practical side, the new velocities are easy to implement and require minimal changes to any code already based on MPDATA. Besides the new expressions for the antidiffusive velocities, the only additional change is that the new velocities must be properly bounded to insure stability and positive definiteness of the solution.

APPENDIX A

Theorem A: Suppose the variables $U^{(k)}$ and $V^{(k)}$ are defined by the recursion

$$\begin{aligned} U^{(k)} &= \left|U^{(k-1}\right| (1 - \left|U^{(k-1}\right|) A - 2 (F) U^{(k-1} V^{(k-1)} B \\ V^{(k)} &= \left|V^{(k-1)}\right| (1 - \left|V^{(k-1)}\right|) B - 2 (1 - F) U^{(k-1} V^{(k-1)} A \end{aligned} \qquad \text{(A-1)}$$

Suppose that $0 < A < 1$ and also $0 < B < 1$. Then

a. There always exists a choice of the constant F such that

$$\begin{aligned} \text{sign} (U^{(k)}) &= \text{sign } A \quad k = 1,2,3, \ldots \\ \text{sign} (V^{(k)}) &= \text{sign } B \quad k = 1,2,3, \ldots \end{aligned} \qquad \text{(A-2)}$$

b. This choice of F may always be made within the range [0 , 1].

c. When F lies within this range, then

$$\begin{aligned} \Sigma_U &\equiv \sum_{k=1}^{\infty} U^{(k)} \quad \text{is convergent} \\ &\qquad \text{and} \\ \Sigma_V &\equiv \sum_{k=1}^{\infty} V^{(k)} \quad \text{is convergent} \end{aligned} \qquad \text{(A-3)}$$

Proof:

a. Since we have assumed that both A and B are positive, it is necessary to prove that $U^{(k)}$ and $V^{(k)}$ are also positive for all k. We prove this by induction. Begin by using the Courant condition (5.1) to eliminate $V^{(k)}$ from the first recursion. This leads to an inequality

$$U^{(k)} \geq [\left|U^{k-1}\right| - (U^{k-1})^2][A - 2 B F] \qquad \text{(A-4)}$$

Similarly in the recursion for $V^{(k)}$, use (5.1) to eliminate $U^{(k-1)}$. This yields the inequality

$$V^{(k)} \geq [\left|V^{k-1}\right| - (V^{k-1})^2][B - 2 A(1 - F)] \qquad \text{(A-5)}$$

Assume now that $U^{(k-1)} > 0$ and also that $V^{(k-1)} > 0$. From (5.4) it follows that if

$$F \leq \frac{A}{2B} \qquad \text{(A-6)}$$

then $U^{(k)} > 0$. Also, if we choose

$$\frac{2A - B}{2A} \leq F \qquad \text{(A-7)}$$

then $V^{(k)} > 0$.

It remains to show that there always exists an F that simultaneously satisfies (5.6) and (5.7). This is equivalent to showing that the following inequality always holds for A and B positive:

$$\frac{2A - B}{2A} \leq \frac{A}{2B} \qquad \text{(A-8)}$$

But (5.8) can be written in the form

$$0 \leq \frac{(A - B)^2}{2AB} \qquad \text{(A-8)}$$

which is always true.

Thus there always exists a value of F that satisfies (A-6) and (A-7). From (4.9), we have $U^{(1)} > 0$ and $V^{(1)} > 0$. Also, we have shown that if $U^{(k-1)} > 0$ and also $V^{(k-1)} > 0$, then $U^{(k)} > 0$ and $V^{(k)} > 0$. By induction, the first part of the theorem is proven.

b. We prove that we can always choose F from the range [0, 1] by construction. In particular, the choice

$$F = \min\left\{\frac{A}{2B}, \max\left[0, 1 - \frac{B}{2A}\right]\right\} \qquad \text{(A-9)}$$

satisfies both (A-6) and (A-7). Other choices of F are possible. In general, we note that (A-6) never requires a negative value of F, nor does (A-7) ever require F to exceed 1.

c. Finally, we prove that choosing a value of F from the range [0 , 1] insures convergence of the sum. Returning to the recursion relations, we note that

$$\begin{aligned} 2(F)|U^{k-1}||V^{k-1}|B &\geq 0 \\ 2(1 - F)|U^{k-1}||V^{k-1}|A &\geq 0 \end{aligned} \qquad \text{(A-10)}$$

Thus we can write

$$U^{(k)} \leq \left|U^{k-1}\right| (1 - \left|U^{k-1}\right|) A$$
$$V^{(k)} \leq \left|V^{k-1}\right| (1 - \left|V^{k-1}\right|) B \qquad \text{(A-11)}$$

It is then easy to show that $U^{(k)} < 1$ and $V^{(k)} < 1$, based on proof by induction. Finally, we apply the limit test

$$\lim_{k\to\infty} \left|\frac{U^{(k)}}{U^{(k-1)}}\right| < A < 1$$
$$\lim_{k\to\infty} \frac{\left|V^{(k)}\right|}{\left|V^{k-1}\right|} < B < 1 \qquad \text{(A-12)}$$

to prove convergence of the sum Σ_U and Σ_V.

QED

* * *

REFERENCES

[1] P.K. Smolarkiewicz, "A Fully Multidimensional Positive Definite Advection Transport Algorithm with Small Implicit Diffusion," J. Comp. Phys. **54**, p.325 (1984).

[2] P.K. Smolarkiewicz and T.L. Clark, "The Multidimensional Positive Definite Advection Transport Algorithm: Further Development and Applications," J. Comp. Phys. **67**, p.396 (1986).

[3] P. K. Smolarkiewicz and W. W. Grabowski, "The Multidimensional Positive Definite Advection Transport Algorithm: Nonoscillatory Option," to be published in the J. Comp. Phys. **86** (1990).

[4] A. Harten, "High Resolution Schemes for Hyperbolic Conservation Laws", J. Comp. Phys. **49**, p357 (1983).

[5] S. K. Gudonov, Math. Sb. **47**, p.271 (1959). also: Cornell Aero. Lab. Translation.

[6] W. Kaplan, Advanced Calculus, 3rd Edition, Addison Wesley, Menlo Park, 1984, p.385.